

# 구조적 상이성 분석에 기반한 XML 문서 변환 시스템의 설계 및 구현

조 정 길<sup>†</sup>·조 윤 기<sup>††</sup>·구 연 설<sup>†††</sup>

## 요 약

본 논문은 논리적으로는 유사하지만 구문 측면에서는 서로 다른 XML 스키마(Schema) 기반의 XML 문서를 구조적 상이성 분석을 통하여 상호 변환하는 시스템의 설계 및 구현에 관한 것이다. 이를 위해 원시(Source) 문서와 목적(Destination) 문서를 데이터 레지스트리(Data Registry)와 구조적 상이성 분석을 이용하여 병합 데이터를 생성하고, 생성된 병합 데이터를 기반으로 원하는 XML 문서를 생성한다. 이 XML 문서 변환 시스템은 다른 응용 시스템에서 사용하는 XML 문서를 현 시스템에 맞게 변환하는 작업을 시간과 비용 그리고 신뢰성의 측면에서 유용하도록 설계하였다. 이 시스템의 구현환경은 IBM 호환 PC에서 동작하며, Windows 2000 환경의 운영체제에서 Visual Basic 6.0을 사용하여 개발하였다.

## Design and Implementation of XML Document Transformation System based on Structured Differences Analysis

Jung-Gil Cho<sup>†</sup> · Youn-Ki Cho<sup>††</sup> · Yeon-Seol Koo<sup>†††</sup>

## ABSTRACT

This paper handles the design and implementation of the system for transforming the XML document based on XML Schema being different in syntax but similar in logic, with using structured differences analysis. In the system, the merge data is generated from the source and destination documents by utilizing data registry and structured differences analysis, and then XML document is generated from the generated merge data. The XML document transformation system is designed that transformation process to the present application system from the different application system gains advantage in the aspect of time, cost, and reliability. The implementation environment of the system is that it is run on IBM compatible PC and it is developed using the software of Visual basic 6.0 with the platform of Windows 2000.

키워드 : XML, XML 스키마, XML 문서 변환, 구조적 상이성 분석(structured differences analysis), 데이터 레지스트리(data registry)

### 1. 서 론

오늘날, 인터넷은 넓은 범위의 영역으로부터 데이터를 교환하는 거대한 수의 공급자와 수요자가 거주하는 글로벌 장터로 볼 수가 있다[1]. 따라서 인터넷 문서 표준인 XML을 기반으로 한 전자 문서의 교환과 처리가 다양한 시스템에 응용되고 사용범위가 점차 넓어지고 있다. 이에 따라 각 분야에서 특성에 맞는 XML 문서의 표준이 정의되고 있다. 그러나 특정 단체나 플랫폼에 맞추어 정의된 XML 문서 표준은 각기 다른 구조를 가지고 있기 때문에 원시 문서와 목적 문서 간의 효율적인 문서의 교환과 처리를 위해서는 서로 연결해 줄 수 있는 문서 변환 처리가 요구된다. 또한, 웹 응용 프로그램

웹에 XML을 사용하는 이유는 간결성(Simplicity), 풍부한 자료구조, 국제적 문자에 대한 탁월한 처리의 장점을 가지고 있기 때문이다[2]. 그런데 웹을 기반으로 하는 응용 시스템 운영에 있어서 가장 큰 문제는 응용 시스템간의 데이터 공유이다[3]. 각 응용 시스템에서 사용하는 데이터를 시스템에 맞게 변환하는 작업은 많은 시간과 비용 그리고 신뢰성의 측면에서 시스템 운영자나 개발자에게 큰 부담이 되고 있다.

XML 스키마는 문서 구조에 대한 규칙을 정의한다. 그러나 XML 문서에 있는 태그가 나타내는 의미를 표현하는 규칙은 정의되어 있지 않기 때문에 유사한 내용의 태그를 식별하기란 불가능하다. 이에 데이터 레지스트리를 기반으로 태그 및 속성을 XML에서 사용하면 XML 스키마를 공유하지 않고도 XML 문서의 자동화된 변환을 할 수가 있다[4,5]. 즉 본 논문에서 XML 문서 변환은 서로 다른 응용 시스템들이 XML 스키마에서 정의한 태그 및 속성의 의미를 알고 있음

† 정 회 원 : 남서울대학교 컴퓨터학과 겸임교수  
†† 준 회 원 : 주성대학 인터넷가상현실과 교수  
††† 정 회 원 : 충북대학교 컴퓨터과학과 교수  
논문접수 : 2001년 9월 10일, 심사완료 : 2001년 12월 6일

을 가정한다.

따라서 본 논문은 문서의 구조적 정보를 정의하는 XML 스키마를 통하여 데이터를 제공하는 원시 문서 측과 데이터를 처리하는 목적 문서 측의 구조를 가지고 원하는 형태의 구조관계인 병합 데이터를 생성한다. 생성된 병합 데이터를 기반으로 원하는 XML 문서를 생성하는 XML 문서 변환 시스템을 설계 및 구현하였다.

본 논문에서 제안하는 XML 문서의 변환 방법은 데이터 처리를 필요로 하는 목적 문서 측의 구조에 맞게 데이터가 재구성 되도록 변환해 준다. 이러한 변환처리는 확장성과 데이터의 상호 운용성을 높게 되어 목적 문서 측에서 원하는 형태로 의미부여가 가능해진다.

본 논문의 구성은 다음과 같다. 2장에서는 시스템의 이해와 개발에 필요한 관련연구에 대하여 설명하고, 3장에서는 XML 문서 변환 방법에 대하여 제시한다. 4장에서는 구현 및 고찰에 대하여 기술하고, 5장에서는 결론 및 향후 연구 과제에 대하여 논의한다.

## 2. 관련 연구

### 2.1 XML 스키마

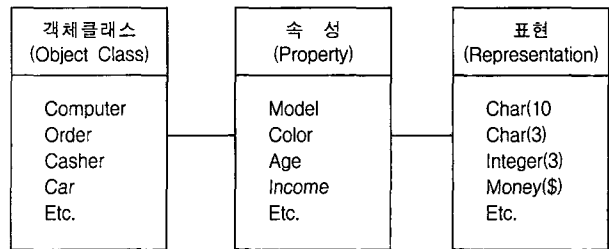
XML 문서에서 SGML(Standard Generalized Markup Language)처럼 DTD(Document Type Definition)를 기본으로 하여 유효한 문서만을 허락하지 않고 적격문서도 허락한 이유는 모든 XML 문서를 한번 읽을 때마다 문서의 유효성을 검증하는 데 시간을 소비하는 것을 방지하고, 개인적으로 마크업을 자유롭게 구성하여 XML 문서를 만드는 것을 지원하기 위해서이다. 그러나 정확한 구조를 예상하는 XML 문서 응용 분야에서는 신뢰성과 안정성이 고려되어야 하기 때문에 특정 단체나 플랫폼과 정보를 공유하기 위해서는 문서의 구성 방법과 마크업 방법을 정확히 규정해야 한다. 따라서 적격문서보다 더 엄격한 조건을 적용한 문서를 스키마 문서라 한다. 이러한 스키마 문서를 작성하는 규칙들을 규정한 정의에는 DTD와 스키마가 있다.

XML 스키마는 XML 문서의 내용을 제안하고 기술하는 XML 언어이며, 현재 W3C 권고안 단계(Recommendations phase)로 발표하였다[6-8]. XML DTD는 XML과 다른 문법을 사용하고, 이름공간(Namespace)을 지원하지 못하며, 데이터형이 제한적이며, 복잡하고 느슨한 확장 메카니즘을 가지고 있는 등의 뚜렷한 한계를 가지고 있다[9, 10]. 이에 XML 스키마는 DTD의 단점을 보완하기 위해 만들어졌다. 이러한 DTD의 단점을 보완하게 위한 특징들은 45가지의 다양한 데이터형을 지원하고, 사용자 정의 데이터형을 선언할 수가 있으며, 이름공간을 지원하고, 사용자 정의 데이터형이나 상속을 재정의 하며, 속성을 그룹핑하고, 객체 지향적이고 집합을 표현할 수가 있다.

### 2.2 데이터 레지스트리

데이터 레지스트리는 데이터의 특성에 대한 사실들을 저장하는 저장소이며, 이러한 데이터에 대한 유일한 식별 및 등록 등의 역할을 한다. 데이터 레지스트리는 시스템 또는 조직 사이의 데이터 공유를 지원하며, 공유 데이터의 사용자로 하여금 데이터의 의미, 표현, 식별 등의 대한 일반적인 이해를 가능하게 한다.

데이터 레지스트리에 저장되는 데이터를 기술하는 단위를 데이터 요소라고 한다[4, 5]. 데이터 요소는 ISO/IEC 11179에서 규정하였으며, 객체 클래스(Object Class), 속성(Property), 표현(Representation)의 세 부분으로 구성된다. (그림 1)은 데이터 요소의 구조를 보여준다.



(그림 1) 데이터 요소의 구조

객체 클래스는 수집하거나 저장하려는 데이터를 지칭하며, Computer, Order, Cashier, Car 등이 있다. 속성은 사물을 설명하거나 구별하는데 사용되는 특성이며, Model, Color, Age, Income 등이 있다. 표현은 해당 데이터 요소가 가지는 값에 대한 도메인 및 표현 형태를 기술하며, Char, Integer, Money 등으로 구분한다.

데이터 요소는 유일한 식별자(Identifier)를 가진다. 데이터 레지스트리를 사용하는 모든 사용자나 프로그램이 이 식별자 값에 위해서 데이터 요소를 구별해 낼 수가 있다. 데이터 요소는 식별자 외에 데이터의 정의, 이름, 문맥 등이 기술된다. 객체 클래스의 각 속성은 하나의 데이터 요소로 대응된다. 각각의 데이터 요소의 예는 <표 1>과 같다.

<표 1> 데이터 요소의 예

식별자	데이터요소 이름	표현
Cah001	Casher Name	Char(10)
Cah002	Casher Age	Integer(3)
...	...	...
Car001	Car Name	Char(20)
Car002	Car Color	RGB color
...	...	...

### 2.3 XML 문서 변환

논리적으로는 유사하지만 구문 측면에서는 서로 다른 XML 문서를 상호 변환하는 데에는 여러 가지 방법이 있다. 현재

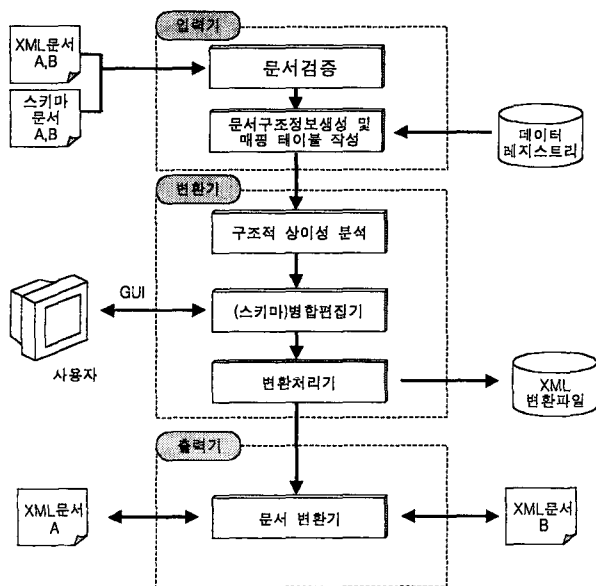
이러한 방법의 연구는 대략 3가지 종류로 이루어지고 있다. XML 스키마 기반의 XML 문서를 목적 문서에 맞게 XSL 스타일시트로 생성하여 처리하는 XSL 스타일시트 생성 방법[11], XML DTD 기반의 XML 문서를 목적 스키마에 맞게 문서구조를 변환하여 사용할 수 있는 변환기를 생성하는 변환기를 이용한 XML 문서의 상호 변환 방법[12], XML DTD 기반의 XML 문서를 데이터 레지스트리와 구조적 상이성을 이용하여 목적 스키마에 맞게 문서를 변환하는 메타데이터 레지스트리를 이용한 XML 문서의 상호 변환 방법[13] 등이 있다.

현재 연구되고 있는 변환 방법들을 비교 분석하여 보면 XSL 스타일시트 생성 방법과 변환기를 이용한 XML 문서의 상호 변환 방법은 사용자가 변환정보내용을 전부 입력해야 한다. 또한 메타데이터 레지스트리를 이용한 XML 문서의 상호 변환 방법은 데이터 레지스트리를 사용하여 유사한 내용의 태그를 식별 가능하게 되어 구조적 상이성을 분석하여 부분 자동화가 가능하나 구조적 상이성 분석 구분이 미흡하다.

### 3. XML 문서 변환 방법

본 논문에서는 XML 문서 변환 방법에 대하여 (그림 2)과 같이 제안한다. 이러한 XML 문서 변환 방법은 다음과 같은 두 가지 방법의 시나리오가 필요하다[2]. 첫째, 원시 문서 측에서 생성된 XML 문서(원시 문서 스키마를 토대로 함)가 목적 문서 측의 스키마를 적용해도 유효하도록 변환한다. 둘째, 목적 문서 측에서 생성된 XML 문서(목적 문서 스키마를 토대로 함)가 원시 문서 측의 스키마를 적용해도 유효하도록 변환한다.

#### 3.1 시스템 구성



(그림 2) 시스템 전체 구성도

본 논문에서 설계한 시스템은 크게 세 부분으로 구분한다. XML 문서와 스키마를 입력받아 각 문서의 구조를 생성하는 입력기, 양측 요소간에 호환이 가능하도록 XML 문서를 변환하는 변환기, 완성된 변환기 코드를 가지고 수신자 쪽의 스키마에 맞게 XML 문서를 변환하여 출력하는 출력기로 나뉜다. (그림 2)는 이 시스템의 전체 구성도를 나타낸다.

#### 3.1.1 입력기

입력기 단계에서의 문서 검증은 데이터 제공 측의 XML과 스키마 문서와 데이터를 처리하는 측의 XML과 스키마 문서를 입력으로 받는다. 이 단계는 문서 검증기를 이용하여 XML 문서와 XML 스키마 문서의 유효성을 검사하고 XML 스키마를 추출한다. 문서검증 단계에서의 파싱(Parsing)은 하나의 XML 문서를 읽은 후에, 이를 나중에 분석할 수 있는 구성요소(element)를 나누는 과정을 말한다. 이러한 XML 문서 파싱(구문 분석)은 XML 프로세서(processor)를 사용한다. 이 시스템에서 사용한 검증기(Parser)는 Microsoft사의 MSXML Version 3.0을 사용하였다. 이 검증기는 XML 프로세서로서 일종의 유효성 검사를 담당한다[14].

문서검증 단계에서 사용된 검증기는 입력된 XML 문서를 추상적으로 구조화하며, 문서구조정보인 스키마 트리를 생성한다. 이 스키마 트리는 DOM level 1에 기초한 구조이며, 문서 요소를 구조화하기 때문에 접근이 쉽고 문서처리를 가능하게 한다. DOM의 추상적인 구조는 사용자가 볼 수가 없는 프로그램상의 처리를 돕는 형태이다. 이에 DOM 구조를 사용자가 볼 수 있는 형태의 구조로 변경한다.

문서구조정보는 유효한 2개의 XML 스키마를 가지고 생성하며 구조적 상이성 분석에서 사용한다. 매핑 테이블은 XML 스키마에서 식별자를 가지고 데이터 레지스트리를 참조하여 생성한다.

#### 3.1.2 변환기

구조적 상이성 분석은 문서 변환시에 고려할 구조적 상이성에 근거하여 문서를 변환한다. 변환과정은 이 논문에서 제안한 7가지의 구조적 상이성의 경우를 반복적(recursive)으로 적용한다. 저장된 문서구조정보와 두 문서의 매핑 테이블을 이용하고 구조적 상이성 분석을 통하여 변환 트리를 생성한다. 이때 발생하는 예외 상황 처리는 병합 편집기의 GUI 환경에서 사용자가 변환 트리를 완성하여 병합 데이터를 생성한다.

병합 편집기는 사용자가 GUI 환경에서 각각의 스키마 문서 정보를 통해 변환 데이터를 완성하는 단계이다. 이 단계에서는 사용자가 쉽고 편리하게 원하는 형식의 변환을 할 수 있도록 매핑에 필요한 세부적인 정보들과 여러 형태의 매핑 인터페이스를 제공한다. 구조적 상이성 분석에서 처리하지 못한 예외상황이 발생했을 경우에는 이 단계에서 사용자가 적절한 처리를 하도록 한다. 이러한 처리를 완료하면 변환처리

의 기준이 되는 병합 데이터를 완성하게 된다.

변환 처리기는 두 문서간의 매핑 정보를 포함한 병합 데이터를 이용하여 XML 변환 파일(XML Transformation File, XTF)을 만든다. XTF는 XML 매핑을 위한 파일로서 하나의 XML 문서를 다른 문서로 변환한다.

3.1.3 출력기

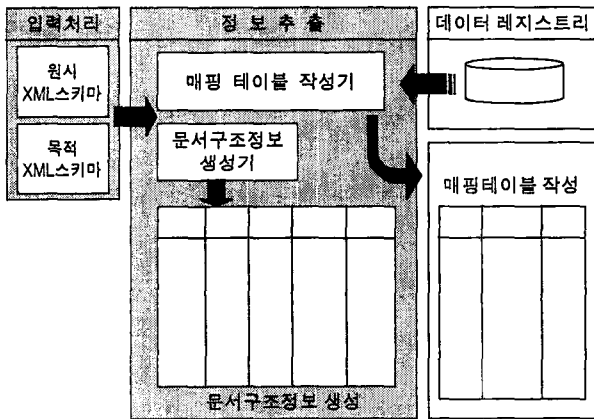
문서 변환기는 XTF를 입력으로 받아 하나의 XML 문서를 다른 문서로 변환한다. 원시 문서 쪽의 XML 문서를 목적 문서 쪽의 XML 스키마에 맞게 변환한다. 동일한 규칙 파일이 역방향의 변환에도 그대로 적용된다.

3.2 변환 방법

3.2.1 문서구조정보 생성 및 매핑 테이블 작성

문서구조정보 생성기는 유효한 2개의 XML 스키마를 입력으로 받아서 문서구조정보를 생성한다. 이 문서구조정보 데이터는 구조적 상이성 분석에서 이용된다.

(그림 3)는 문서 입력값을 원시 XML 스키마와 목적 XML 스키마로 받아서 문서구조정보 생성기가 문서구조 정보를 생성하고, 매핑 테이블 작성기가 매핑 테이블을 작성하는 과정을 나타낸 것이다. 이때 매핑 테이블을 작성하기 위해서는 데이터 레지스트리가 이용된다.



(그림 3) 문서구조정보 생성 및 매핑 테이블 작성기

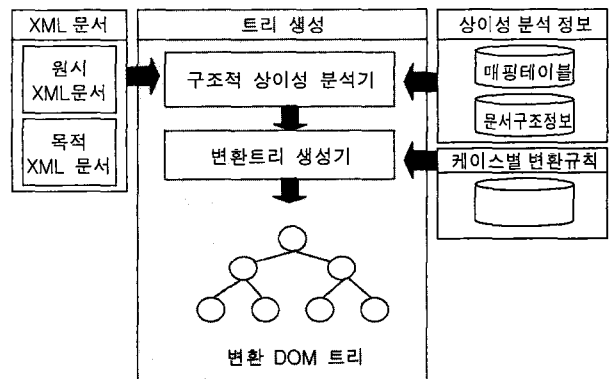
문서구조정보의 데이터 형식은 "(부모노드 계층정보, 현재 노드 계층정보, 현재노드 종류, 현재 노드명, 자식노드 순서 정보)"로 표현한다. 각각의 항목은 구성요소(element)의 부모, 자식간의 계층정보와 현재 노드의 상태정보를 나타낸다. 부모노드 계층정보, 현재노드 계층정보, 자식노드 순서정보는 /number 형태의 변수로 나타낸다. 문서의 루트 노드는 노드 값으로 /1을 가지며, 루트 노드의 부모노드는 /0 값을 가진다. 현재노드 종류는 문서의 구성요소를 1, 데이터 값을 2, 속성 값을 3으로 표기한다. 현재 노드명은 노드의 이름을 표기하며, 자식노드 순서정보는 자식노드 계층정보의 변수를 차

별로 표기한다. 현재노드종류에 데이터 값이 오는 경우는 자식노드가 없으므로 NULL로 표기한다.

매핑 테이블 작성은 XML 스키마에서 식별자를 가지고 있는 태그들을 테이블 형태로 작성하며, 같은 태그들을 구별한다. 매핑 테이블의 표현정보는 유효한 2개의 XML 스키마를 데이터 레지스트리를 참조하여 생성하며 매핑 테이블의 데이터 형식은 "(구성요소 식별자, 원시문서의 구성요소 타입, 목적문서의 구성요소 타입)"로 표현한다.

3.2.2 구조적 상이성 분석

구조적 상이성 분석기는 원시 XML 문서와 목적 XML 문서를 입력으로 받아서 문서 변환시에 고려해야할 구조적 상이성에 근거하여 문서를 변환한다. 변환 과정은 본 논문에서 제안한 7가지의 구조적 상이성의 경우를 반복적(recursive)으로 적용한다. 저장된 문서구조정보 데이터와 두 XML 문서의 매핑 테이블을 이용하고, 구조적 상이성 분석을 통하여 변환 트리를 생성한다. 이때 발생하는 예외 상황 처리는 병합 편집기의 GUI 환경에서 사용자가 변환 트리를 완성하여 병합 데이터를 생성한다. 다음의 (그림 4)는 구조적 상이성 분석을 통한 변환 트리 생성기의 세부 구성도이다.



(그림 4) 구조적 상이성 분석을 통한 변환 트리 생성기

구조적 상이성 분석기에서 XML 문서 변환 시에 고려해야할 구조적 상이성을 7가지로 자세히 세분화하여 다음과 같이 제시하였다.

(1) 노드명이 일치하지 않는 경우

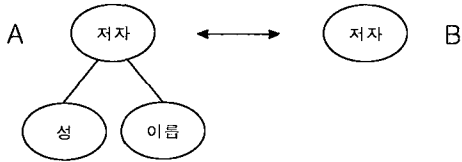
노드의 구조가 1 : 1 관계로 서로 대응되는 경우이다. 즉 양쪽 노드의 노드명이 같은 의미를 나타내지만 <paper>와 <논문>으로 틀리게 표현되는 경우이다.



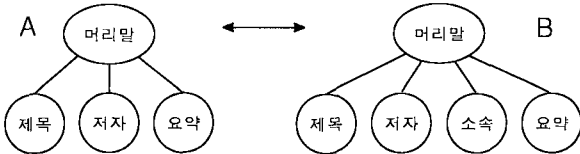
(2) 한쪽에만 포함 관계가 있는 경우

한쪽의 경우에만 자식 노드가 존재하는 경우이다. 위의 그림에서와 같이 한쪽에는 부모 노드 <저자> 밑에 자식 노드

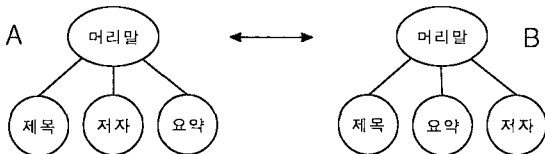
<성> <이름>이 존재하고 다른 쪽에는 자식 노드가 없는 <저자> 노드만 있는 경우이다.



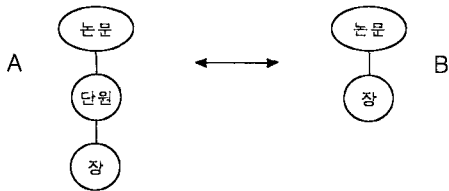
(3) 자식 노드의 개수가 틀린 경우  
양쪽의 부모 노드에 딸린 자식노드의 개수가 틀린 경우이다.



(4) 자식 노드의 순서가 틀린 경우  
양쪽의 부모 노드에 딸린 자식 노드의 개수가 틀린 경우이다.

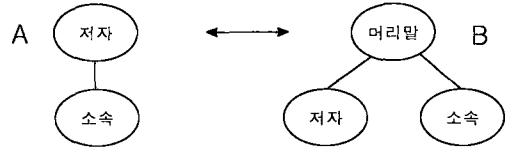


(5) 노드가 추가되거나 생략된 경우  
한쪽 노드의 구조가 <논문> <단원> <장>으로 이루어져 있으나, 다른쪽 노드의 구조는 <단원> 노드가 생략되고 <논문> <장>으로 이루어진 경우이다.

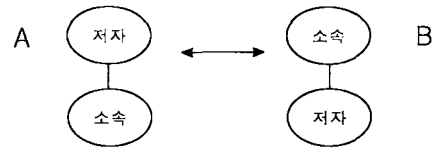


(6) 수평-수직 관계  
한쪽의 계층 구조에는 부모 노드와 자식 노드의 관계가 다

른쪽의 계층 구조에서는 같은 레벨의 노드로 표현되는 경우이다.



(7) 포함 관계가 반대인 경우  
한쪽의 계층 구조에는 부모 노드와 자식 노드의 포함 관계가 다른쪽의 계층 구조에서는 포함 관계가 반대로 표현되는 경우이다.



(8) 예외 상황  
위에서 7가지로 구분하여 명시되지 않은 구조적 상이성이 발생한 경우이거나, 노드 구조의 변경이 불가능한 경우이다. 이때 발생하는 예외 상황 처리는 사용자가 미완성의 변환 트리를 병합 편집기의 GUI 환경에서 완성하여 병합 데이터를 생성한다.

본 논문에서는 앞에서 제안한 구조적 상이성에 대한 케이스별 해결방법을 다음의 <표 2>로 해결한다.  
변환 과정은 제안한 7가지의 경우를 <표 2>에 제시한 해결 방법에 따라 반복적으로 적용한다. 먼저, 변환할 문서를 노드명이 일치하지 않는 경우에 대하여 태그 변환을 시행한다. 다음에, 한쪽에만 포함관계가 있는 경우에 해당하는 것을 찾아서 태그변환을 실시한다. 이처럼 일치시키는 과정을 (3), (4), (5), (6), (7)의 경우에도 적용되며, 이 과정을 반복적으로 실행한다. (8)에서 언급한 예외 상황이 발생하였을 경우에는 GUI에서 사용자의 적절한 변환처리가 이루어지도록 한다. 이러한 변환처리 과정을 거치게되면 병합 데이터를 생성하게 된다.

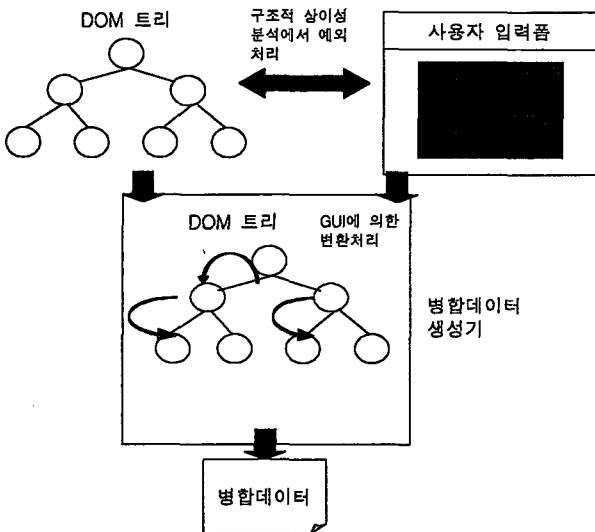
<표 2> 구조적 상이성에 대한 케이스별 해결 방법

구조적 상이성 관계	A → B	B → A
노드명이 일치하지 않는 경우	같은 식별자를 가지는 태그로 교체	같은 식별자를 가지는 태그로 교체
한쪽에만 포함관계가 있는 경우	서브태그의 식별자가 없는 경우 값을 합치고 태그 삭제	예외상황 발생시킴
자식 노드의 개수가 틀린 경우	대칭되는 태그외에 예외상황 발생시킴	대칭되는 태그외에 예외상황 발생시킴
자식 노드의 순서가 틀린 경우	B에 맞게 태그의 순서를 조정	A에 맞게 태그의 순서를 조정
노드가 추가되거나 생략된 경우	생략되어야할 태그 삭제	추가해야할 태그를 첨가함
수평-수직 관계	하위태그를 없애고 상위태그와 동일한 레벨로 태그 옮김	하위태그에 속한 태그를 없애고 하위태그를 첨가하여 태그 옮김
포함관계가 반대인 경우	태그의 위치를 서로 변경. 서브태그의 값은 상위태그의 속성값으로 추가	태그의 위치를 서로 변경. 서브태그의 값은 상위태그의 속성값으로 추가
예외상황	예외상황에 대한 메시지출력, 사용자가 처리	예외상황에 대한 메시지출력, 사용자가 처리

### 3.3 병합 편집기

구조적 상이성 분석에서 예외 상황이 발생한 경우에 생성된 병합 데이터는 사용자가 임의로 처리하는 부분이 있기 때문에 완전한 변환 DOM 트리가 되지 못한다. 이 단계의 GUI 환경에서 사용자가 내용에 맞게 처리를 하여 완전한 변환 DOM 트리를 완성하게 된다.

문서구조정보와 매핑 테이블 내용을 참조하여 생성한 병합 데이터의 형식은 "(구조적 상이성 분석관계, 원시문서노드 계층정보, 목적문서노드 계층정보, 원시문서 서브구조, 목적문서 서브구조)"로 표현한다. 구조적 상이성 분석 관계 항목은 구조적 상이성 분석의 7가지 중에 속하는 번호를 기입하며, 원시문서노드 계층정보와 목적문서노드 계층정보 항목은 해당하는 노드의 번호를 기입한다. 또한 원시문서 서브구조와 목적문서 서브구조 항목은 앞의 항목에 딸린 내용을 기입한다. (그림 5)는 병합 편집기가 병합 데이터를 생성하는 과정을 나타낸 것으로서, 사용자 입력폼에서 구조적 상이성 분석에서 예외 처리된 부분을 사용자가 GUI에 의한 변환처리 과정을 통하여 완성된 DOM 트리를 가지고 병합데이터 생성기가 병합 데이터를 생성시킨다.



(그림 5) 병합 편집기

### 3.4 변환 처리기

변환 처리기는 전 단계에서 완성된 병합 데이터를 입력으로 XTF를 생성한다. XTF는 XML 언어 형태로 이루어져 있기 때문에 XML과 같은 특징을 갖게된다. 이러한 XTF의 사양은 다음의 (그림 6)과 같다.

XTF의 루트 구성요소(element)는 `xtf : rules`이다. 이 루트 구성요소는 한개 이상의 `xtf : trans` 구성요소를 자식으로 갖는다. 하나의 `xtf : trans` 구성요소는 하나의 변환 규칙을 나타내며, 변환 규칙의 원시 문서와 목적 문서에 각각 해당하는 `xtf : str`과 `xtf : dtr` 구성요소를 자식으로 갖는다. `xtf : str`인

`from`-변환은 원시 문서의 부분과 일치하는데 사용되며, `xtf : dtr`인 `to`-변환은 목적 문서를 구축하는데 사용된다. 반대 방향으로 변환하고자 할 경우에는 `from`-변환과 `to`-변환의 역할을 바꾸어서 적용하면 된다.

```

<xtf : rules>
  <xtf : trans>
    <xtf : str> ..... </xtf : str>
    <xtf : dtr> ..... </xtf : dtr>
  </xtf : trans>
  <xtf : trans>
    <xtf : str> ..... </xtf : str>
    <xtf : dtr> ..... </xtf : dtr>
  </xtf : trans>
  :
  :
</xtf : rules>
    
```

(그림 6) XTF의 사양

변환 패턴의 `from`-변환과 `to`-변환은 원시 문서의 일부와 일치하는 문서의 서브구조를 가지고 있다. 변환 패턴이 임의의 요소와 일치하면 `/number` 라는 변수가 나타난다. 이 변수안에 있는 숫자는 1부터 시작하며, 두 변수가 동일한 숫자를 가질 수는 없다. 제한사항으로서 XTF 변환 패턴인 구성요소 `xtf : str` 과 `xtf : dtr`은 각각 한 개의 구성요소만을 가질 수가 있다.

## 4. 구현 및 고찰

### 4.1 구현

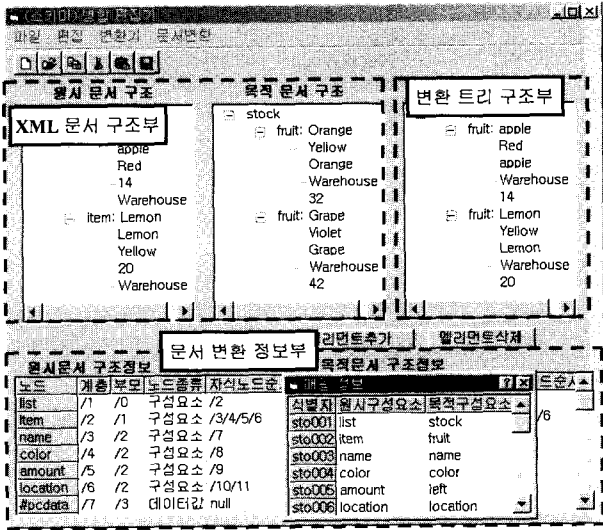
본 논문에서 개발한 시스템의 구현환경은 IBM PC 호환 컴퓨터에서 동작하며, Windows 98의 운영체제 환경 하에서 비주얼 툴인 Visual Basic 6.0을 사용하여 개발하였다. 검증기는 마이크로소프트사의 COM으로 구성된 MSXML 3.0을 사용하였다[14]. 이 검증기는 프로세서로서 일종의 유효성 검사를 담당한다.

#### 4.1.1 사용자 인터페이스 설계

본 시스템의 사용자 인터페이스(UI) 구조는 문서의 생성 및 사용자가 환경 지원을 위해 마우스 드래그 앤 드롭(Drag and Drop) 기능으로 구현하였다. 본 시스템은 크게 XML 문서 구조부, 문서변환 정보부, 변환 트리 구조부로 구성되었으며, 시스템의 사용자 인터페이스는 다음의 (그림 7)과 같다.

XML 문서 구조부의 원시문서 구조부와 목적문서 구조부는 읽어들이 XML 문서를 사용자가 쉽게 파악할 수 있도록 트리 구조를 나타낸다. 문서변환 정보부는 문서 변환에 필요한 원시문서 구조정보와 목적문서 구조정보 그리고 단축 메뉴를 이용하여 매핑 정보를 보여주며, 예외사항 처리에 대한 변환 DOM 트리 완성에 참고한다. 변환 트리 구조부는 완성

된 변환 DOM 트리를 보여준다.



(그림 7) 사용자 인터페이스

4.1.2 문서구조정보 생성 및 매핑 테이블 작성

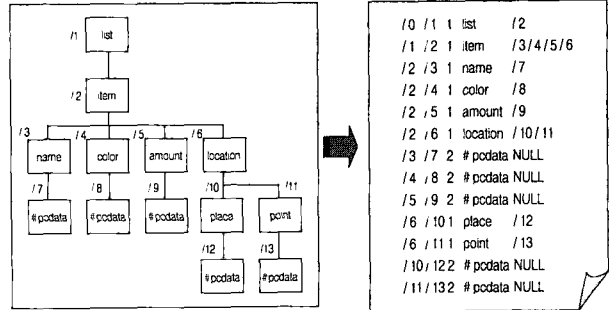
문서구조 정보 생성은 입력된 원시 XML 스키마와 목적 XML 스키마를 기반으로 하였으며, 매핑 테이블 작성은 입력된 원시/목적 XML 스키마와 데이터 레지스트리를 이용하여 생성하였다. <표 3>은 논리적으로는 유사하지만 구문 측면에서는 서로 다른 구조를 가지는 두개의 재고 XML 문서를 나타내며, 이 재고 XML 문서를 이용하여 (그림 8)과 같은 문서구조 정보와 데이터 레지스트리를 이용하여 (그림 9)와 같은 매핑 테이블을 생성한다.

<표 3> 두 재고 XML 문서의 비교

Stock-A.xml	Stock-B.xml
<pre>&lt;list&gt; &lt;item&gt;   &lt;name&gt;apple&lt;/name&gt;   &lt;color&gt;Red&lt;/color&gt;   &lt;amount&gt;14&lt;/amount&gt;   &lt;location&gt;&lt;place&gt;Warehouse   &lt;/place&gt;     &lt;point&gt;A&lt;/point&gt;   &lt;/location&gt; &lt;/item&gt; &lt;item&gt;   &lt;name&gt;Lemon&lt;/name&gt;   &lt;color&gt;Yellow&lt;/color&gt;   &lt;amount&gt;20&lt;/amount&gt;   &lt;location&gt;&lt;place&gt;Warehouse   &lt;/place&gt;     &lt;point&gt;C&lt;/point&gt;   &lt;/location&gt; &lt;/item&gt; &lt;/list&gt;</pre>	<pre>&lt;stock&gt; &lt;fruit&gt;   &lt;color&gt;Yellow&lt;/color&gt;   &lt;name&gt;Orange&lt;/name&gt;   &lt;location&gt;Warehouse X   &lt;/location&gt;   &lt;left&gt;32&lt;/left&gt; &lt;/fruit&gt; &lt;fruit&gt;   &lt;color&gt;Violet&lt;/color&gt;   &lt;name&gt;Grape&lt;/name&gt;   &lt;location&gt;Warehouse Y   &lt;/location&gt;   &lt;left&gt;42&lt;/left&gt; &lt;/fruit&gt; &lt;/stock&gt;</pre>

<표 3>의 두 재고 문서중 Stock-A.xml 문서의 XML 스

키마를 트리 구조로 표현하고, 그것을 문서구조 정보로 생성되는 내용은 다음의 (그림 8)과 같다.



(그림 8) 트리 구조와 문서구조 정보

두 재고 문서의 XML 스키마에서 식별자를 가지고 있는 태그들을 데이터 레지스트리를 이용하여 매핑 테이블로 작성되는 결과는 다음의 (그림 9)와 같다. (그림 9)에서 구성요소 식별자 sto001에 있는 list 와 stock와 sto002에 있는 item과 fruit는 의미는 같고 표현만 다른 태그가 된다. 이러한 정보는 데이터 레지스트리를 참조한다.

sto001	list	stock
sto002	item	fruit
sto003	name	name
sto004	color	color
sto005	amount	left
sto006	location	location
sto007	place	NULL
sto008	point	NULL

(그림 9) 구성요소 식별자에 따른 매핑 테이블

4.1.3 병합 편집기

1	/1	/1	null	null
1,4	/2	/2	/3/4/5/6	/4/3/6/5
null	/3	/4	null	null
null	/4	/3	null	null
null	/5	/6	null	null
2	/6	/5	/10/11	/9
null	/7	/8	null	null
null	/8	/7	null	null
null	/9	/10	null	null
null	/10	b	null	null
null	/11	b	null	null
null	/12	/9	null	null
null	/13	/9	null	null

(그림 10) 병합 데이터의 내용

구조적 상이성 분석은 원시 XML 문서와 목적 XML 문서

를 입력으로 받아서 문서 변환시에 고려해야할 구조적 상이성 분석 정보를 이용하여 문서를 변환한다. 이러한 변환 과정은 본 논문에서 제안한 7가지의 구조적 상이성의 경우를 반복적(recursive)으로 적용하여 변환 DOM 트리를 생성한다. 구조적 상이성 분석기와 변환트리 생성기를 통해 생성된 변환 DOM 트리를 병합 편집기에 보내고 GUI에 의한 예외상황 처리를 거쳐서 병합 데이터를 생성하게 되는데, 작성된 병합 데이터의 내용은 (그림 10)와 같다.

4.1.4 XML 변환 파일 생성 및 변환 결과

병합 편집기에서 생성한 병합 데이터인 (그림 10)를 두 문서간의 변환에 사용되는 XTF로 변환하면 다음의 (그림 11)과 같다. XTF인 stock.xtf는 두 재고 문서들간의 변환에 사용된다.

```

<?xml version="1.0"?>
<xtf:rules>
  <xtf:trans>
    <xtf:str><list>/1;</list></xtf:str>
    <xtf:dtr><stock>/1;</stock></xtf:dtr>
  </xtf:trans>
  <xtf:trans>
    <xtf:str><item><name>/3;</name><color>/4;</color><amount>/5;</amount>
      <location><place>/10;</place><point>/11;</point></location></item>
    </xtf:str>
    <xtf:dtr><fruit><color>/4;</color><name>/3;</name><location>/10;/11;</location>
      <left>/5;</left></fruit>
    </xtf:dtr>
  </xtf:trans>
  <xtf:trans>
    <xtf:str><#pcdata>/12;/13;</#pcdata></xtf:str>
    <xtf:dtr><#pcdata>/9;</#pcdata></xtf:dtr>
  </xtf:trans>
</xtf:rules>
    
```

(그림 11) 재고 XML 문서 stock.xml의 XTF인 stock.xtf

앞의 사용자 인터페이스에서 변환 작업에 따라 완성된 병합 데이터는 주 윈도우 화면의 메뉴에서 변환처리를 통해 XTF를 생성하며, 이 XTF는 메뉴에 있는 출력기의 문서 변환기에 의해 원하는 결과를 얻게된다. 다음의 (그림 12)은 원시 XML 문서를 목적 XML 스키마에 맞게 변환된 결과를 보여준다.

```

- <stock>
- <fruit>
  <color>Red</color>
  <name>apple</name>
  <location>Warehouse A</location>
  <left>14</left>
</fruit>
- <fruit>
  <color>Yellow</color>
  <name>Lemon</name>
  <location>Warehouse C</location>
  <left>20</left>
</fruit>
</stock>
    
```

노드	개	원시문서
list	/1	
item	/2	
name	/3	
color	/4	
amount	/5	구성요소 /9
location	/6	구성요소 /10/11
#pcdata	/7	데이터값 null

(그림 12) 변환 결과

4.2 고찰

본 논문은 데이터를 제공하는 원시 문서 추과 데이터를 처리하는 목적 문서 추의 구조를 가지고 문서의 구조적 정보를 정의하는 XML 스키마를 통하여 문서를 변환하는 XML 문서 변환 시스템의 설계와 구현에 관한 연구이다. 본 시스템에서 사용한 문서 검증기는 검증기의 인터페이스만을 이용하여 문서를 검증하므로 개선되어 버전이 향상된 검증기도 지원이 가능하다. 또한 XML 스키마 기반의 XML 문서를 데이터 레지스트리를 사용하고 구조적 상이성 분석을 7가지로 구별하여 사용자가 예외 상황에 대해서만 최소한의 처리를 하게 하였으며, 변환기의 형식을 XML 문서 형식으로 생성하여 서로의 구조적 정보에 유효하면서도 한쪽의 구조에 종속되지 않는 독립적인 플랫폼 형태의 처리가 가능하도록 하였다. 그러나 데이터 레지스트리를 이용해서 XML 문서를 생성해야만 이러한 자동화된 문서 변환을 할 수가 있다.

본 시스템의 장점은 구조적 상이성 분석을 통한 변환의 수작업 처리를 최소화하였으며, 시스템을 모듈화하여 부분적인 수정 및 대체가 가능하도록 하였다.

미흡한 점으로는 XTF 변환 패턴 안에 하나의 구성요소만을 수 있는 제한사항이 있으며, Windows 운영체제 환경 하

<표 4> 타 변환 시스템과의 비교

시 스템	특징 및 장점	단 점
메타데이터 레지스트리를 이용한 XML 문서의 상호 변환[13]	데이터 레지스트리를 사용하여 유사한 내용의 태그 식별 가능, 구조적 상이성을 분석하여 부분 자동화	구조적 상이성 구분이 미흡, 데이터 레지스트리 사용의 제약조건
변환기를 이용한 XML 문서의 상호 변환 [12]	시스템 구조를 컴포넌트화, 변환기 생성으로 상호변환이 쉬움	사용자가 변환정보내용 전부 입력
XSL 스타일 시트 생성[11]	확장성이 좋음	문서 구조정보의 변환 없이 XSL만 생성, 사용자가 변환정보내용을 전부 입력
본 시스템	데이터 레지스트리를 사용하여 유사한 내용의 태그 식별 가능, 세분한 구조적 상이성 분석을 통한 수작업 최소화, 모듈화하여 부분적인 수정 및 대체가 가능	데이터 레지스트리 사용의 제약조건, 플랫폼에 종속적



에서 Visual Basic 6.0을 사용하여 개발하여서 플랫폼에 종속적이다.

본 논문에 관련된 연구에서 문서 변환 시에 고려해야할 구조적 상이성을 [2]에서는 (1), (2), (4)의 3 가지로 구분하였고, [13]에서는 (1), (2), (6), (7)의 4가지로 구분하였다. 그러나 이러한 구분에서는 변환시에 처리해야할 요소가 부족한 실정이다. [2]에서는 자식노드의 갯수가 틀린 경우, 노드가 추가되거나 생략된 경우, 수평-수직 관계, 포함 관계가 반대인 경우에 구별을 할 수가 없으며, [13]에서는 자식 노드의 순서나 갯수가 틀린 경우, 노드가 추가되거나 생략된 경우에 구별을 할 수가 없다. 이에 본 논문에서는 더 자세히 7가지로 세분화하여 문서 변환 시에 고려해야할 구조적 상이성을 구분하여 제시한다.

본 논문의 문서 변환 시스템에 대한 연구는 국내외에서 개발되어 상용화된 시스템이 아직 없으며, 연구를 진행하고 있는 추세이다. <표 4>는 타 변환 시스템과 본 시스템의 특징 및 장점, 단점을 비교하였다.

## 5. 결론 및 향후 연구

인터넷 환경에서의 문서 처리 및 이 기종 시스템간의 정보 교환이 날로 증대되어 가고 사용자의 요구사항은 점점 다양해져 가고 있다. 이에 문서표준인 XML은 21세기 인터넷과 e-Business를 이끌어 갈 핵심 기술로 자리잡아가고 있다.

최근에 XML을 기반으로 한 다양한 웹 애플리케이션에서 비즈니스 문서 등과 같이 논리적으로는 유사하지만 구문 측면에서는 서로 다른 구조들로 이루어진 문서들이 증가하고 있다. 이러한 환경에서의 문서교환과 처리를 위해서는 서로의 구조적 정보에 유효하면서 한쪽의 구조에 종속되지 않는 독립적인 플랫폼 형태의 문서 변환기가 필요하다.

이에 본 논문에서는 논리적으로는 유사하지만 구문 측면에서는 서로 다른 구조들로 이루어진 XML 스키마 기반의 문서들을 양측간의 원하는 형태의 변환규칙을 데이터 레지스트리를 기반으로 구조적 상이성 분석을 통하여 분석하고, 이를 토대로 XML 변환 파일(XTF)를 생성하여 변환기에 사용하였다. 본 시스템은 원시 XML 문서를 가지고 다른 응용 애플리케이션에서 사용이 가능한 정보를 생성하는데 목적이 있다. 본 시스템에서는 검증된 XML 스키마 문서를 가지고 변환에 필요한 변환기 데이터를 세분화된 구조적 상이성 분석으로 자동화하였고, 예외상황에 대해서는 사용자 선택 방식의 자동 생성으로 사용자 입력이 용이하도록 하였다. 본 시스템의 결과인 XTF는 XML 형식으로 구성되어 있어서 플랫폼에 독립적으로 유용하게 사용될 수 있으리라 본다.

향후 연구과제는 실질적인 전자상거래 시스템에서 처리되는 타 시스템과의 연동작업이 가능하도록 시스템의 확장이 이루어져야 할 것이며, 더욱 세분화된 구조적 상이성 분석이 가능하도록 연구가 필요하다.

## 참 고 문 헌

- [1] Christof Bornhvd, "Semantic Metadata for the Integration of Web-based Data for Electronic Commerce," IEEE, Nov., 1999.
- [2] Hiroshi Maruyama, Kent and Naohiko Uramoto, "XML and Java Developing Web Applications," Addison Wesley Longman, 1999.
- [3] Rik Drummond, Kay Spearman, "XML Set to Change the Face of E-Commerce," Network Computing, Vol.9, No.8, pp.140-144, May, 1998.
- [4] "Information technology - Specification and standardization of data element," ISO/IEC 11179-1 Final Committe Draft, June, 1998.
- [5] 나홍석, 채진석, 김창화, 백두권, "차세대 웹 상에서의 문서 교환 및 검색을 위한 프레임워크", 정보처리, 제6권 제3호, pp. 52-61, 1999.
- [6] W3C, "XML Schema Part 0 : Primer," <http://www.w3.org/TR/xmlschema-0>, 2, May, 2001.
- [7] W3C, "XML Schema Part 1 : Structures," <http://www.w3.org/TR/xmlschema-1>, 2, May, 2001.
- [8] W3C, "XML Schema Part 2 : Datatypes," <http://www.w3.org/TR/xmlschema-2>, 2, May, 2001.
- [9] Eric van der Vlist, "Using W3C XML Schema," <http://www.xml.com>, 2000.
- [10] Norman Walsh, "Understanding XML Schemas," <http://www.xml.com>, 2000.
- [11] 성길용, 강치원, 정희경, "구조적 문서 변환을 위한 XML Mapper 시스템 설계 및 구현", 정보과학회 춘계학술발표논문집, 2001.
- [12] 심민석, 유대승, 엄정섭, 강만모, 이명재, "XTGen : XML 변환기 생성을 위한 컴포넌트 기반 시스템", 정보과학회 춘계학술발표논문집, 2001.
- [13] 홍종하, 양유승, 나홍석, 백두권, "메타데이터 레지스트리를 이용한 XML-문서 교환 방법", 정보과학회 춘계학술발표논문집, 2001.
- [14] Microsoft, "MSXML Parser 3.0 Release," <http://msdn.microsoft.com/>, Apr., 2001.
- [15] Mark Colan, "The application of XSL for XML transformation in e-business solutions," XML EUROPE 2000, 2000.
- [16] Mikael Peltier, Francois Ziserman, Jean Bezivin, "On levels of model transformation," XML EUROPE 2000, 2000.
- [17] David Burdett, "Effective strategies for integrating businesses on the Net using XML," XML EUROPE 2000, 2000.
- [18] Alex Homer "XML IE5 Programmer's Reference," Wrox Press, 1999.
- [19] 김형도, "B2B 전자상거래@XML", 배움터, 2000.
- [20] 이증호, "XML과 전자상거래", 정보문화사, 2001.
- [21] Simon North, Paul Hermans, "Teach Yourself XML," SAMS, 1999.
- [22] W3C, "Extensible Markup Language(XML) Version 1.0," <http://www.w3.org/TR/REC-xml>, Feb., 10, 1998.



**조 정 길**

e-mail : innocom@dreamwiz.com  
1987년 숭실대학교 전자계산학과 졸업(학사)  
1993년 숭실대학교 정보과학대학원 정보산업  
학과(이학석사)  
2002년 충북대학교 대학원 전자계산학과 박  
사과정 수료

1996년~현재 남서울대학교 컴퓨터학과 겸임교수  
관심분야 : 객체지향 자료 모델링, XML 문서관리, 질의처리, 정보  
검색



**조 윤 기**

e-mail : ykcho@selab.chungbuk.ac.kr  
1994년 충북대학교 컴퓨터과학과 졸업(학사)  
1996년 충북대학교 대학원 전자계산학과  
졸업(이학석사)  
2002년 충북대학교 대학원 전자계산학과  
(이학박사)

2002년~현재 주성대학 인터넷가상현실과 교수  
관심분야 : 전자상거래, 정보 시각화, XML 문서 관리, 정보 검색



**구 연 설**

e-mail : yskoo@cbucc.chungbuk.ac.kr  
1964년 청주대학교 졸업(학사)  
1975년 상군관대학교 경영행정대학원 전자  
자료처리학과(경영학석사)  
1981년 동국대학교 대학원 통계학과(이학  
석사)

1988년 광운대학교 대학원 전자계산학과(이학박사)  
1994년~1995년 한국정보과학회 부회장  
1979년~현재 충북대학교 컴퓨터과학과 교수  
관심분야 : 객체지향 테스트, 품질관리, 정보 검색, 전자상거래